Hierarchical Material Recognition from Local Appearance

Supplementary Material

S1. The Dearth of Material Image Data

While material recognition as a line of research has existed for over 60 years, there exist only a few dozen widely available datasets. This is compared to thousands of datasets targeted at object recognition, segmentation, *etc*., which can be more easily aggregated from online images and annotated by the layperson without any additional information. These two factors make the task of creating material image datasets very challenging. Nevertheless, there exists a need in the space of material recognition for not only more, but much larger datasets.

Beyond material recognition, there is rich literature on adjacent tasks that are highly correlated with recognizing materials. Several prior works introduce datasets for texture recognition [4], material segmentation [1, 2, 12], texture segmentation [16], and estimation of the bidirectional texture function (BTF) or bidirectional reflectance function (BRDF) [5, 15, 18]. Even so, such datasets are often limited in their range of material categories. For example, while BRDF/BTF datasets may contain tens of thousands of images spread across a reasonable range of classes (usually between 10 to 40), typically, each class only has a few unique physical instances per class. Compare this to ImageNet [6] for object recognition, which contains 1,000 classes with equally many instances per class. As a result, material recognition datasets have limited specificity with such few classes, and given the limited intraclass diversity, learned models may not generalize well to real-world tasks. We contribute *Matador*, introduced in Sec. 4 and further detailed in Sec. S2, as a step towards overcoming this hurdle.

S2. Matador Dataset Details

A subset of the most recent material recognition datasets used as benchmarks in this work is presented in Tab. S1. We compare our *Matador* dataset and the subset *Matador-C1* (see Sec. S2.5) to these datasets. We find *Matador* contains higher interclass and intraclass diversity compared to existing datasets, and additionally incorporates new datatypes. The dataset is available on the *Matador* webpage, and representative samples are shown in Fig. S1.

S2.1. Acquisition Details

Here we provide further details about how the *Matador* dataset was constructed. As discussed in Sec. 4, we collected in-the-wild material images to populate the taxonomy with image data. We used an Apple iPhone 15 Pro Max to do this, of which we utilized three of its sensors: the wide-angle camera, lidar (which is registered with the wide-angle camera), and ultrawide-angle camera. We first take a close-up image of the material's local appearance using the wide-angle camera. Simultaneously, a lidar scan of the material's local surface structure is captured. We then take a step back to capture the material in the context of its surroundings using the ultrawide-angle camera. Between these two captures, we record the motion of the phone using the inertial measurement unit (IMU). The complete sample of a material thus contains the following raw data:

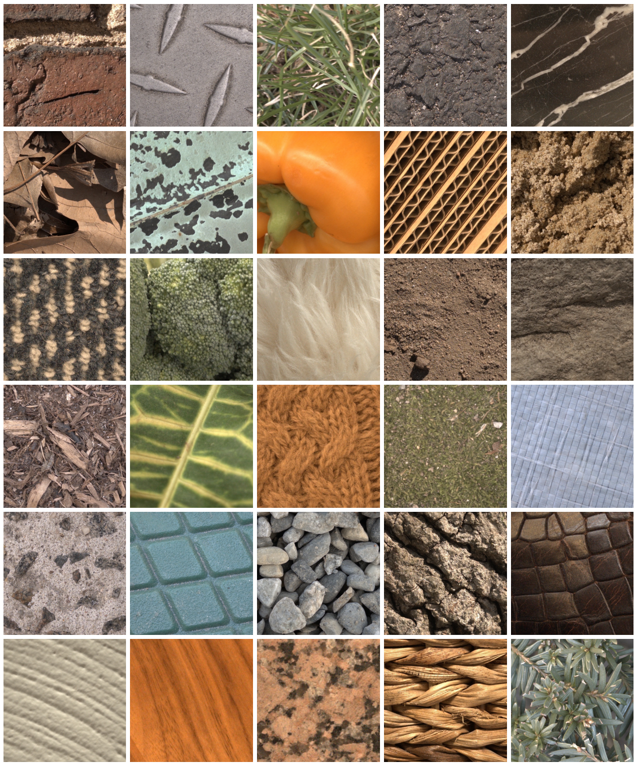


Figure S1. ***Matador*** **Local Appearance Examples.** Several examples of the local appearance images in the *Matador* dataset. These images are captured at a close distance (typically around 20-30 cm away), and "local appearance" denotes that the target material fills the majority of the frame. More examples can be examined using the interactive viewer on the *Matador* webpage.

1. **Local Appearance Image.** The image is 12 MP (3024×4032 px^2) with a 74° FOV from the wide-angle camera. Since the 48 MP wide-angle camera used to capture local appearance has a Quad-Bayer color filter array (for high dynamic range imaging in a single shot), the resolution after capture is reduced to 12 MP. The image is recorded as a 12-bit Bayer raw DNG.
2. **Depth Map of the Surface Structure.** The image is a 100 points/degree^2 lidar depth map registered with the local appearance image. It has an equivalent FOV to the

Table S1. Statistics on Datasets Used in This Work. We compare the size and diversity of each benchmark dataset with the *Matador* dataset, which contains a high number of material classes and instances per class (interclass and intraclass diversity). We refer the interested reader to Liu et al. [13] for metrics on older and adjacent datasets. Note that "*Matador* (raw)" denotes the unprocessed version of the dataset, with images that are neither undistorted, demosaiced, nor cropped to a relevant region of interest.

Dataset	Authors	Year	Classes	Images	Resolution [px^2]	Avg. Instances per Class	Image Datatypes
KTH-TIPS-2b	Caputo et al. [3]	2005	11	4,752	200×200	4	Appearance
FMD	Sharan et al. [17]	2013	10	1,000	512×384	100	Appearance
GTOS	Xue et al. [19]	2017	40	34,243	240×240	∼15	Appearance, Angular
GTOS-Mobile	Xue et al. [20]	2018	31	6,066	455×256	∼5	Appearance
Matador	Ours	2025	57	7,238	512×512	∼126	Appearance, Depth, Context
Matador-C1	Ours	2025	37	6,614	512×512	∼179	Appearance, Depth, Context
Matador (raw)	Ours	2025	57	7,238	3024×4032	∼126	Appearance, Depth, Context

local appearance image, but when projected into two dimensions, the depth map is roughly a fifth of the resolution. Thus, it must be resampled to approximate per-pixel depth in the local appearance. The depth map is recorded as a 32-bit float binary file.

3. **Global Context Image.** The image is 12 MP (3024×4032 px^2) with a 104° FOV from the ultrawide-angle camera. It is recorded as a 12-bit Bayer raw DNG.
4. **Motion.** The IMU data contains accelerometer measurements in the time between the local appearance image capture and the global context image capture. Measurements are sampled at 100 Hz – the maximum for the iPhone 15 Pro Max. The motion is recorded in a timestamped text file.
5. **Metadata.** The metadata records camera intrinsics and extrinsics (from the wide-angle to ultrawide-angle camera) for each capture, as reported by the manufacturer. In addition, typical EXIF data is also recorded that includes the camera settings for each capture (*e.g.*, ISO, exposure time, and suggested white balance gains).

We record Bayer raw images to eliminate compression artifacts that could alter a material's appearance. The above datatypes represent the raw dataset. We developed an iOS application to complete this capture process; it can also be found on the *Matador* webpage. Note that in Swift, the wide-angle camera and lidar correspond to the `builtInLiDARDepthCamera`, and the ultrawide-angle camera corresponds to the `builtInUltraWideCamera`.

S2.2. Processing Raw Data into Matador

To process the raw data into the *Matador* dataset, we first demosaic and undistort all images. Resampling is applied to the depth map to match the resolution of the local appearance image. We then define an approximately 5×5 cm^2 region of interest (using depth for scale) in the local appearance containing the material being captured and excluding extraneous information. The final appearance image is an sRGB 16-bit unsigned integer TIFF at 512×512 px^2, and the corresponding depth map is a 512×512 px^2 32-bit float TIFF. When resizing images, a Mitchell-Netravali filter is used for enlarging and a Lanczos filter is used for shrinking.[1] The context image remains 3024×4032 px^2 but is now an sRGB 16-bit unsigned integer TIFF. Motion data and metadata are YAMLs. We additionally correct for vignetting using manufacturer-provided gains (see the Adobe DNG specification), which is only noticeable in the full-resolution context images.

S2.3. Formal Treatment of Novel View Rendering

We now formalize the rendering process described in Sec. 4 used to generate novel views from *Matador*, which are then utilized during model training. While this rendering process is standard, its application to material recognition is novel, and we detail the effects of each augmentation parameter below for completeness.

Novel view generation is motivated by the fact that the visual appearance of materials can vary wildly depending on the scale they take on in an image. By generating novel views, a model trained on both the real and rendered datasets may improve in generalization to real-world settings. This is supported by the out-of-distribution (OOD) results in Fig. 5a. A qualitative look at this can be seen in Fig. 6, where materials are recognized at distances well beyond those captured in the local appearance images of *Matador* that are used for training.

Since the 3D structure and position of the material are known, we can alter camera pose and imaging characteristics to simulate images produced by different cameras and viewpoints. We assert the true image irradiance E is well-sampled by the local appearance image. With the local appearance and structure, we create a mesh of the material surface where each face represents the scene radiance L – assuming Lambertian reflectance. Spatial transformations H (magnification and orientation) can then be applied to the mesh to change the pose of the imaged material relative to the observing camera. The novel view irradiance E' is proportional to the scene radiance L, and we retrieve E' through raytracing [11] accounting for occlusions, perspec-

[1] The merits of various resampling filters are discussed at length in the ImageMagick [10] usage guide: https://usage.imagemagick.org/filter/.

tive effects, lens aberrations (*e.g.*, defocus), and attenuation (from the aperture, visibility, *etc.*). The irradiance of a novel view is then:

$$E'_b(\mathbf{x}) = \frac{1}{\text{Area}(P)} \int_{\mathbf{p}\in P} L\left(\mathbf{p} + t(F(\mathbf{x}) - \mathbf{p})\right) \, d\mathbf{p}, \qquad \text{(S1)}$$

where t is the length of the ray, and,

$$F(\mathbf{x}) = \mathbf{c} + \frac{d_f}{||\mathbf{x} - \mathbf{c}||} (\mathbf{x} - \mathbf{c}), \qquad \text{(S2)}$$

maps a sensor coordinate $\mathbf{x}$ to a point on the focal plane relative to the piercing point $\mathbf{c}$. The integral is over each point $\mathbf{p}$ on the effective pupil P with full transmission, equivalent to the aperture under a thin-lens model. Spatially-varying blur is then introduced to the novel view by varying the focus distance d_f and size of the pupil P.

The irradiance in Eq. (S1) is continuous and must be sampled to create an image. We first convolve with a box filter p to average over the active area of each pixel:

$$E'_a(\mathbf{x}) = E'_b(\mathbf{x}) * p(\mathbf{x}). \qquad \text{(S3)}$$

The density of rays being traced is then equivalent to sampling with a pulse train with period Δ:

$$E'_s(\mathbf{x}) = E'_a(\mathbf{x}) \cdot \sum_{\mathbf{k}\in\mathbb{Z}^2} \delta(\mathbf{x} - \Delta\mathbf{k}). \qquad \text{(S4)}$$

We additionally model photon and sensor noise with a zero-mean random variable n [8]:

$$E'_n(\mathbf{x}) = E'_s(\mathbf{x}) + n(\mathbf{x}). \qquad \text{(S5)}$$

Finally, the result is discretized by the pixel pitch T to produce an image:

$$E'[\mathbf{u}] = E'_n(\mathbf{u}T). \qquad \text{(S6)}$$

The resulting views are cropped to fill the entire frame and resized to a standard 512×512 px^2 size. As in Sec. S2.2, when resizing images, a Mitchell-Netravali filter is used for enlarging and a Lanczos filter is used for shrinking.

By altering the pose of the material (with H), the defocus of the lens (with d_f and P), the pixel size p, the sensor noise n, and the pitch T, we can simulate how an arbitrary camera would render the material from any distance and orientation. This process is applied to all raw samples in the *Matador* dataset to obtain a larger and more diverse set of training images. Numerous novel views are rendered per raw sample, faithfully depicting how a given material will look under a wide range of imaging conditions.

S2.4. Generalization Test Set Details

Novel views supplement the captured local appearance images used for model training. In addition to local appearance images used for testing, we additionally create a second evaluation set, which we refer to as the out-of-distribution (OOD) test set. As discussed in Sec. 4, the OOD test set is intended to evaluate the effect of generated novel views on model generalization to real-world imaging conditions (such as further viewpoints, harsh blur, and viewing directions that are not frontoparallel). Here we further detail how this test set was created.

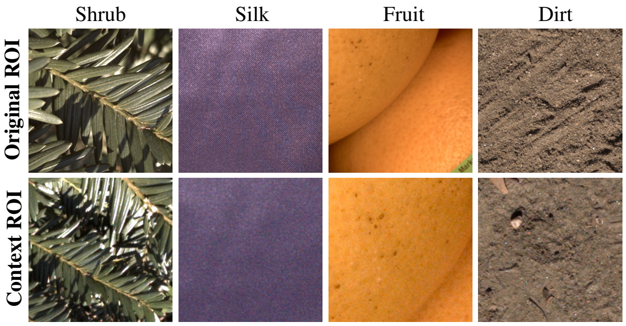
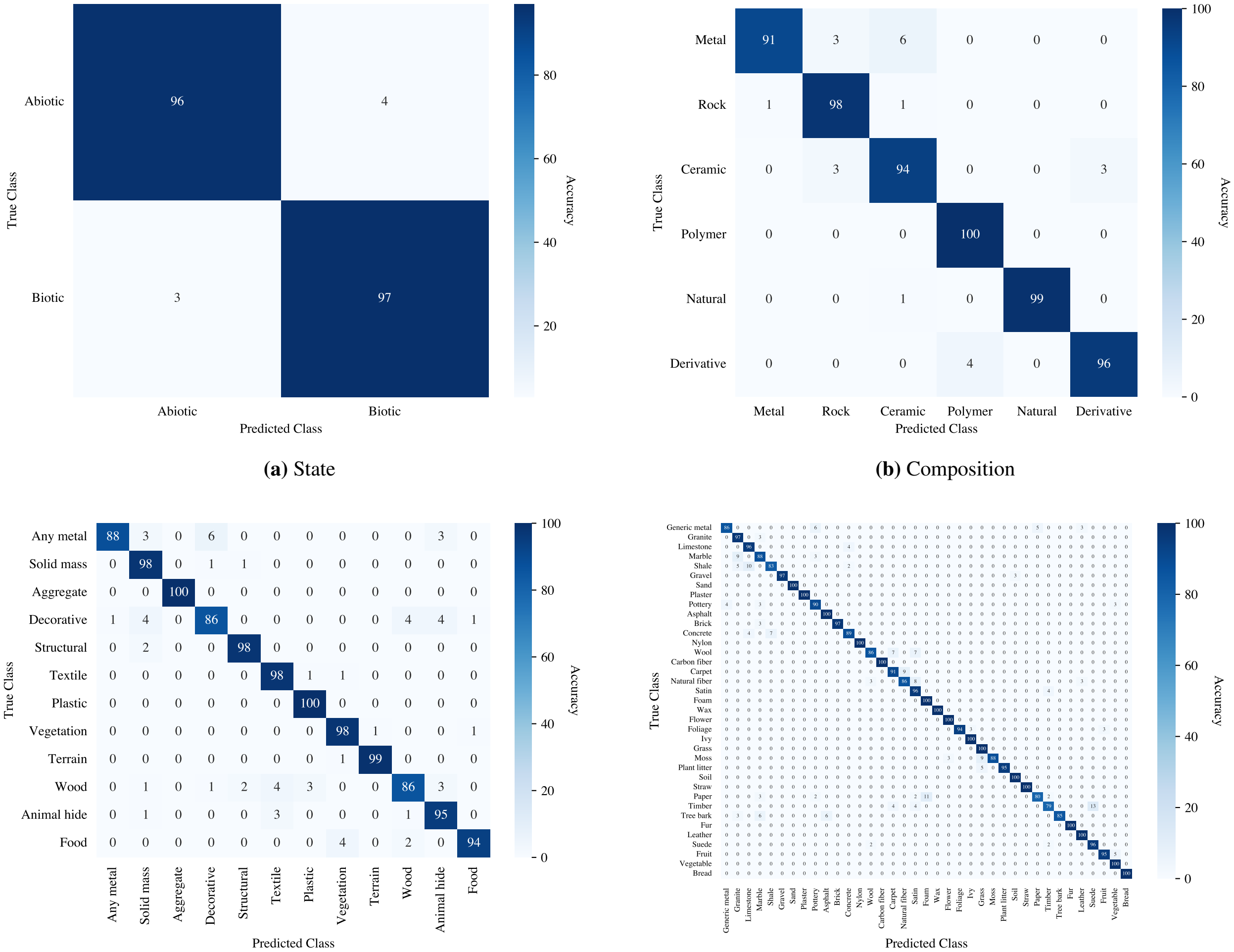


Figure S2. Evaluating Generalization. Examples of the appearance image (top) and patch cropped from the context image (bottom) for *Matador* samples. The latter is used to create an out-of-distribution test set to evaluate real-world model performance (see Sec. S2.4). Compared to the original ROIs used for model training, the context ROIs come from a different camera, a more distant viewpoint, and capture a different area of the material sample. As a result, they generally exhibit reduced detail, have a slight blur, and appear flatter. Zoom in to best see the fine texture differences between the images.

In each raw local appearance image, a square region of interest (ROI) is used to define a relevant patch for model training and testing. This is done to remove any context that might be present in the full-resolution raw local appearance image, and it is typically 5×5 cm^2. Using this same ROI and the registered depth map, we map the four corners of the ROI into three dimensions through unprojection. We then apply a Kalman filter to the accelerometer measurements between the appearance and context captures, and use the resulting spatial transformation to transform the 3D ROI corners into the wide-angle camera's frame of reference at the time of the context image capture. Using manufacturer-provided camera extrinsics, we then transform the 3D ROI corners into the coordinate frame of the ultrawide-angle camera and project them (incorporating differences in camera intrinsics) to get an ROI of similar scale in the context image. This ROI is extracted and resized to 512×512 px^2.

Due to noise in the accelerometer measurements, this is not an exact mapping between the appearance and context ROIs. Instead, the context ROI samples a different location on the target material instance. We manually verify this and that the resulting patches capture the correct material. Since they capture a different location on the material using a more distant viewpoint, different lens, and different image sensor, these patches comprise the OOD test set that simulates realistic real-world capture conditions (see Fig. S2).



(a) State

(b) Composition

(c) Form

(d) Material

Figure S3. Class Accuracy by Taxonomy Level. Top-1 accuracy of our model on *Matador-C1* (Color), per class for each level in our taxonomy. Plots (a-d) are arranged in order of increasing specificity, referring to the levels shown in Fig. 1c. Values are rounded to the nearest integer percentage using the largest remainder method. By framing material recognition as a hierarchical learning problem, we are able to accurately recognize materials at multiple levels of granularity. Note that since all metals are consolidated into a single class in *Matador-C1*, the classes "Any metal" in (c) and "Generic metal" in (d) are equivalent to "Metal" in (b). Nevertheless, we provide additional evaluations at the "Form" and "Material" levels for completeness.

S2.5. Consolidating Matador into Matador-C1

For the performance metrics in Fig. 5a, we consolidated the *Matador* dataset from all 57 categories present in the taxonomy into a smaller set with 37 categories – we refer to the consolidated dataset as *Matador-C1*. In the process, several categories were omitted (thermoplastic, thermoset, elastomer, paint, and glass) that we deemed to have insufficient texture to recognize solely from their local appearance. For instance, glass is featureless and its appearance is dominated by reflections, and paint, being thin, essentially takes on the surface structure of the material it sits on. For similar reasons, we combined other categories into a single one: {aluminum, steel, brass, iron, bronze, copper} are referred to as "generic metal", {stoneware, terracotta, porcelain} as "pottery", {dirt, soil} as "soil", {shrub, foliage} as "foliage", {sandstone, shale} as "shale", {marble, quartz} as "marble", {polyester, silk} as "satin", {cotton, linen} as "natural fiber", {cardboard, paper} as "paper", and {cement, concrete} as "concrete". The visual features of such materials are highly dependent on surface finish and patina, and thus recognition requires knowing context.

For many of the materials omitted or combined in *Matador-C1*, it would be more reasonable to correctly recognize them with the inclusion of global context and even depth images. As the focus of this paper is recognition from local appearance, we did not explore this avenue. However, these data types remain available in the complete *Matador* dataset for future work to leverage.

Table S2. Additional Performance Metrics on *Matador-C1*. Comparison with existing methods. The best overall results are highlighted in bold, and the second best are underlined. Extending Fig. 5a, we add evaluations on grayscale images as well as when only finetuning the classifier head of competing material recognition models. All models use a ResNet50 backbone.

	Top-1 Accuracy ↑		
Method	Grayscale	Color	Out-of-Distribution
Material Recognition Models (Finetuned Classifier Head)			
DeepTEN [21]	67.1	54.0	52.5
DEPNet [20]	69.6	76.8	64.3
FRP [7]	74.2	68.9	58.4
MSLac [14]	74.0	79.1	65.7
Material Recognition Models (Finetuned End-to-End)			
DeepTEN [21]	80.3	88.8	61.5
DEPNet [20]	84.2	87.6	76.1
FRP [7]	84.6	89.4	71.0
MSLac [14]	84.7	88.5	75.4
Hierarchical Material Recognition Models			
Ours	**87.5**	**94.1**	**82.9**

S3. Model Training and Evaluation Details

We now discuss the implementation details of model construction, training, and evaluation. We use a ResNet50 [9] as the image encoder and configure the hierarchical graph attention network defined in Sec. 5 with the following setup:

Input dim: 1024	**Layers:** 2
Hidden dim: 512	**Attention heads:** 1
Output dim: 256	**Pool:** Global Average

Prototype embeddings are then created for each representative class in the dataset with dimensionality equal to the GNN input dimension. The entire model is then trained end-to-end with the following hyperparameters:

Batch size: 400	**LR Schedule:** Cos. anneal.
Epochs: 100	**Weight decay:** 5×10^{-4}
Learning rate: 1×10^{-4}	**Sampling:** Stratified

With a ResNet50 image encoder and the *Matador-C1* classes, the model contains 28.0 M parameters. Training is performed on NVIDIA A6000 Ada GPUs, and if training does not use rendered novel views, it can complete training in under 30 minutes on a single GPU.

We present the performance of our hierarchical graph attention network on *Matador-C1* (Color) for each class in each level of the taxonomy in Fig. S3. Through hierarchical learning, our model exhibits high accuracy at each level. A hierarchical model allows recognition to appropriate levels of specificity depending on the application task, with increasing recognition performance at coarser levels.

In Tab. S2, we present additional evaluations on *Matador-C1*. We evaluate our model and competing methods on color and grayscale images, finetuning competing methods in two ways. In each case, our hierarchical model achieves state-of-the-art classification accuracy.

S4. Material Properties

To enable intelligent systems operating in the physical world, we aggregated a table of mechanical properties (Tab. S3) containing the materials of the taxonomy described in Sec. 3. After visual recognition of a material, the properties present in the table can be used to plan interactions within the environment. For example, if a material is deformable, it will require different handling than if it were rigid. To this end, Tab. S3 is composed of approximate ranges of material density, surface roughness, elasticity, and strength. Its content is retrieved from common textbooks, engineering specification sheets, material databases, and academic publications. These material properties vary in precision – engineered materials (*e.g.*, metals, ceramics, and plastics) are studied extensively and thus their property ranges are well known. For natural materials (*e.g.*, foliage, moss, and grass), rough estimates are provided.

References

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics*, 32(4):1–17, 2013.

[2] Sudong Cai, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. RGB Road Scene Material Segmentation. In *Proceedings of the Asian Conference on Computer Vision*, pages 3051–3067, 2022.

[3] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 1597–1604 Vol. 2, Beijing, China, 2005. IEEE.

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, Columbus, OH, USA, 2014. IEEE.

[5] Kristin J. Dana, Bram Van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] Joao B. Florindo. Fractal pooling: A new strategy for texture recognition using convolutional neural networks. *Expert Systems with Applications*, 243:122978, 2024.

[8] Samuel W. Hasinoff, Fredo Durand, and William T. Freeman. Noise-optimal capture for high dynamic range photography. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 553–560, San Francisco, CA, USA, 2010. IEEE.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceed-*

Table S3. Approximate Mechanical Properties of Common Materials. Sourced from various textbooks, handbooks, and articles on material properties, the present table contains the mechanical properties for each material in our taxonomy. While rough estimates for many categories, such information could be useful to make judgments about whether a recognized material is, *e.g.*, light or heavy, smooth or rough, deformable or rigid, and fragile or strong. Note: a loose approximation of shear strength can also be obtained by multiplying the tensile strength by $1/2$ for brittle materials and $3/5$ for ductile materials.

Material	Density [kg m^{-3}]	Surface Roughness [µm]	Young's Modulus [GPa]	Yield Strength [MPa]	Tensile Strength [MPa]	Poisson's Ratio
Iron (pure/cast forms)	7150–7870	0.1–50	100–210	120–200	130–210	0.26–0.30
Steel (alloyed iron)	7850–8000	0.1–10	190–210	250–1000	400–2000	0.27–0.30
Aluminum (alloys)	2700–2830	0.2–10	69–72	30–400	50–550	0.32–0.35
Brass	8400–8730	0.2–5	100–125	180–250	350–600	0.33–0.36
Bronze	7400–8920	0.2–5	96–120	140–380	240–586	0.34–0.36
Copper	8930–8960	0.1–5	110–130	60–70	200–220	0.34–0.36
Granite	2600–2700	1–50	50–70	–	7–25	0.20–0.30
Limestone	2300–2700	1–50	15–55	–	5–25	0.20–0.30
Marble	2400–2700	0.5–5	50–70	–	7–20	0.20–0.30
Sandstone	2200–2800	5–50	1–20	–	4–25	0.10–0.25
Shale	1770–2670	5–50	1–70	–	2–10	0.10–0.30
Quartz	2600–2700	0.05–5	70–100	–	10–30	0.16–0.18
Dirt (dry, loose)	1000–1600	50–2000	0.01–0.2	–	0–0.02	0.20–0.40
Gravel (loose aggregate)	1500–1800	2000–50000	0.05–0.2	–	0.2–0.6	0.20–0.30
Sand (dry, loose)	1400–1700	50–500	0.01–0.07	–	0.1–0.3	0.20–0.46
Glass (soda-lime)	2500–2600	0.05–1	70–90	–	30–90	0.20–0.25
Plaster (gypsum)	600–1900	1–10	1–10	–	2–6	0.18–0.22
Porcelain	2380–2450	0.1–1	50–74	–	30–50	0.17–0.25
Stoneware	2000–2400	0.5–5	30–70	–	20–40	0.17–0.25
Terracotta (earthenware)	1600–1900	5–50	10–30	–	2–5	0.17–0.25
Asphalt (pavement)	2200–2400	50–500	3–11	–	0.5–3	0.20–0.35
Brick (clay)	1600–2000	5–50	5–20	–	2–5	0.15–0.20
Cement (hardened paste)	1900–2200	5–50	10–30	–	2–5	0.20–0.30
Concrete (aggregate mix)	2200–2500	10–200	25–40	–	2–5	0.10–0.20
Cotton (textile)	1500–1600	0.05–0.5	2–8	–	5–15	0.30–0.40
Linen (textile)	1400–1600	0.1–1	2–8	–	20–60	0.30–0.40
Nylon (textile)	1110–1150	0.002–0.006	2–6	–	200–500	0.35–0.40
Polyester (PET, textile)	1330–1380	0.004–0.006	3–7	–	200–500	0.35–0.40
Silk (textile)	1300–1360	0.005–0.05	5–10	–	100–400	0.30–0.40
Wool (textile)	1200–1400	0.8–3	1–3	–	30–150	0.35–0.45
Carbon Fiber (textile/laminate)	1550–1800	0.007–10	70–500	–	1100–5000	0.20–0.35
Carpet (fiber pile)	100–600	500–5000	0.001–0.01	–	1–3	0.30–0.40
Elastomer (rubber)	900–1200	0.5–50	0.003–0.01	–	5–25	0.48–0.50
Foam (polymer foam)	20–300	0.05–26	0.001–0.1	–	0.05–5	0.10–0.50
Paint (dry film)	1200–1500	0.5–5	0.5–3	–	5–50	0.30–0.40
Thermoplastic (bulk plastic)	900–1400	0.2–5	0.5–3	20–70	30–100	0.32–0.35
Thermoset (rigid polymer)	1100–1500	0.5–5	2–5	30–80	40–80	0.35–0.40
Wax	900–970	1–10	0.03–0.08	–	0.8–1.2	0.30–0.35
Flower	800–1000	0.005–10	0.005–0.02	–	0.2–1	0.30–0.40
Foliage	600–1000	5–25	0.005–0.02	–	0.1–0.3	0.30–0.40
Ivy (vine)	600–1000	5–25	0.1–5	–	5–50	0.30–0.40
Shrub (woody plant)	500–900	5–25	1–10	–	40–100	0.20–0.30
Grass (blade)	800–1000	5–25	0.005–0.02	–	0.1–0.5	0.30–0.40
Moss	50–300	3–10	0.0005–0.005	–	0.1–0.7	0.30–0.40
Plant Litter (dry leaves)	50–100	5–25	0.001–0.005	–	0.01–0.1	0.30–0.40
Soil (moist, packed)	1400–2000	50–2000	0.05–0.2	–	0.02–0.4	0.20–0.40
Straw (aggregate)	70–150	100–1000	1–5	–	20–50	0.30–0.40
Cardboard	600–800	10–100	1–5	–	20–80	0.35–0.45
Paper (sheet)	700–1200	1–20	2–5	–	30–80	0.30–0.40
Timber (wood)	400–900	5–50	6–18	25–140	40–150	0.25–0.40
Tree Bark (outer)	200–600	200–2000	0.01–0.1	–	1–5	0.30–0.40
Fur (pelt)	1200–1400	10–200	1–3	–	30–100	0.35–0.45
Leather	900–1000	5–30	0.2–0.5	–	20–30	0.40–0.45
Suede (split leather)	850–950	10–50	0.1–0.3	–	15–30	0.35–0.45
Fruit (fresh)	900–1100	0.3–15	0.005–0.01	–	0.1–0.3	0.40–0.50
Vegetable (fresh)	800–1100	0.4–14	0.001–0.01	–	0.05–0.5	0.40–0.50
Bread (crumb)	200–400	500–2000	0.00001–0.0002	–	0.001–0.01	0.30–0.40

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[10] ImageMagick Studio LLC. Imagemagick.

[11] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. DR.JIT: A just-in-time compiler for differentiable rendering. *ACM Transactions on Graphics*, 41(4):1–19, 2022.

[12] Yupeng Liang, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. Multimodal Material Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19800–19808, 2022.

[13] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From BoW to CNN: Two Decades of Texture Representation for Texture Classification. *International Journal of Computer Vision*, 127(1):74–109, 2019.

[14] Akshatha Mohan and Joshua Peeples. Lacunarity Pooling Layers for Plant Image Classification using Texture Analysis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5384–5392, Seattle, WA, USA, 2024. IEEE.

[15] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Efficient and Realistic Visualization of Cloth.

[16] Gabriel Schwartz and Ko Nishino. Recognizing Material Properties from Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1981–1995, 2020.

[17] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H. Adelson. Recognizing Materials Using Perceptually Inspired Features. *International Journal of Computer Vision*, 103(3): 348–371, 2013.

[18] Michael Weinmann, Juergen Gall, and Reinhard Klein. Material Classification Based on Training Data Synthesized Using a BTF Database. In *Computer Vision – ECCV 2014*, pages 156–171. Springer International Publishing, Cham, 2014.

[19] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential Angular Imaging for Material Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 764–773, 2017.

[20] Jia Xue, Hang Zhang, and Kristin Dana. Deep Texture Manifold for Ground Terrain Recognition, 2018.

[21] Hang Zhang, Jia Xue, and Kristin Dana. Deep TEN: Texture Encoding Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2896–2905, Honolulu, HI, 2017. IEEE.